GitHub Survey Analysis

Jiaqi Luo

04/08/2017

Analysis Overview

General Survey Infomation

- Sample size
- Sample Demographics
- What do people think of GitHub?

NPS Analysis

- NPS Score
- Margin of Error
- Age /NPS Model

Other Found Out

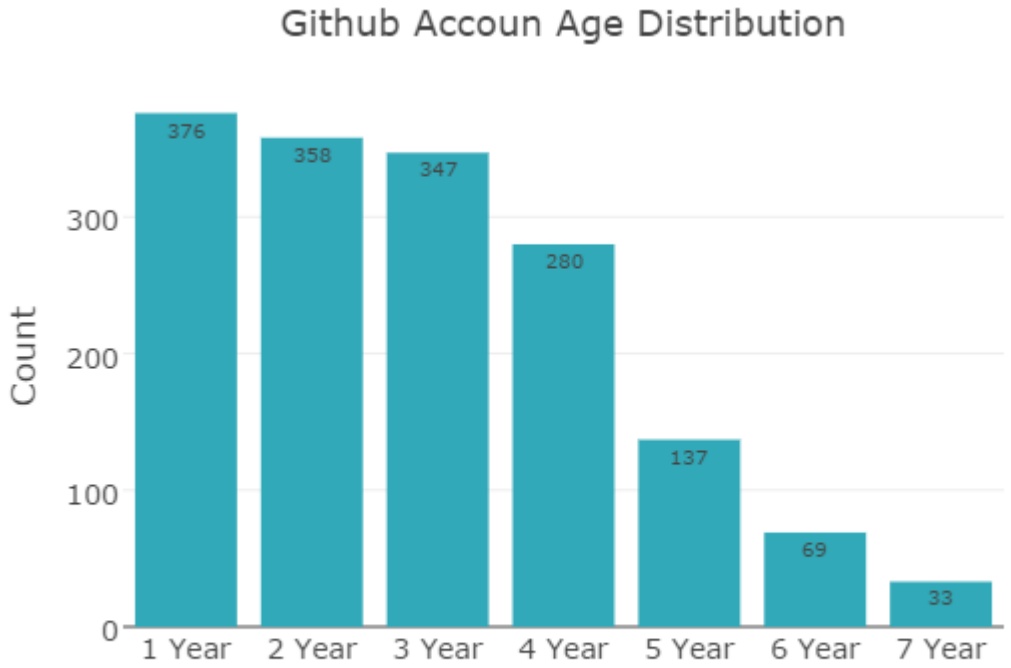
- Role vs. NPS
- Improvement vs. NPS
- Segmentations
- Text Analysis

Future Works

- Summary
- Future Analysis

General Survey Info

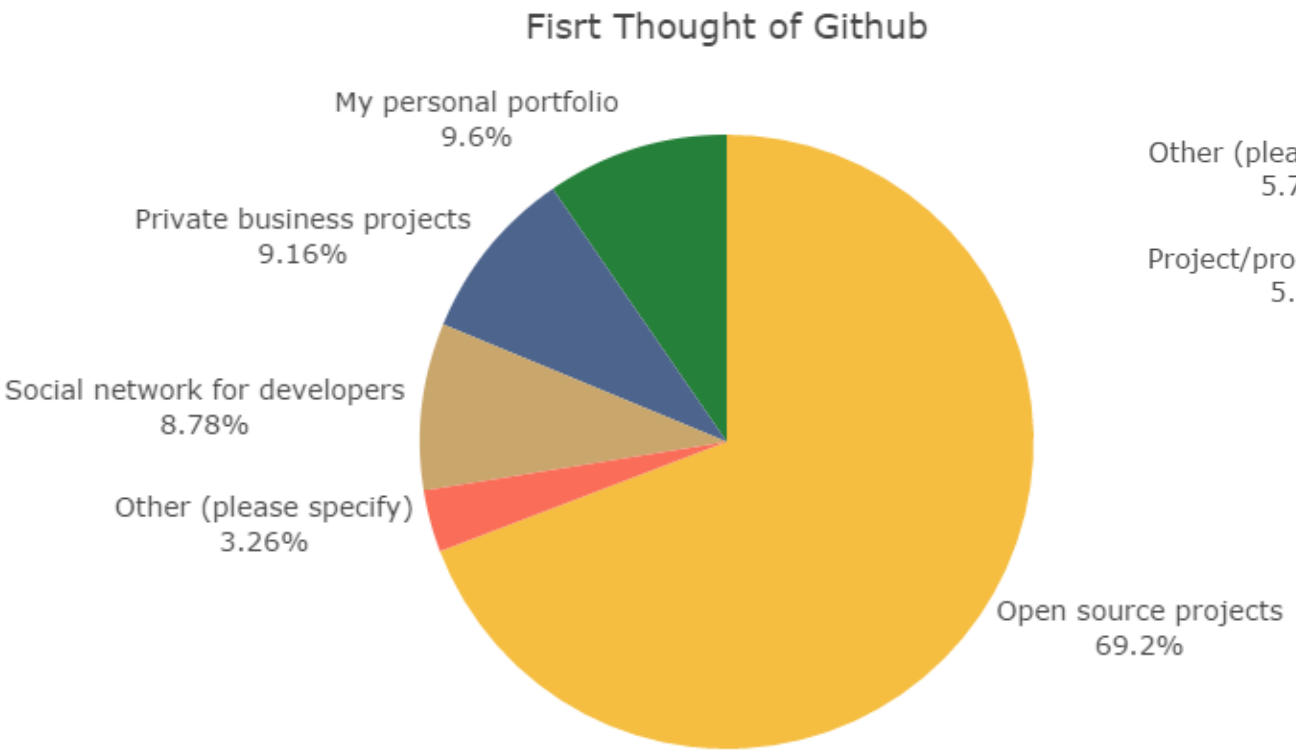
1600
Samples

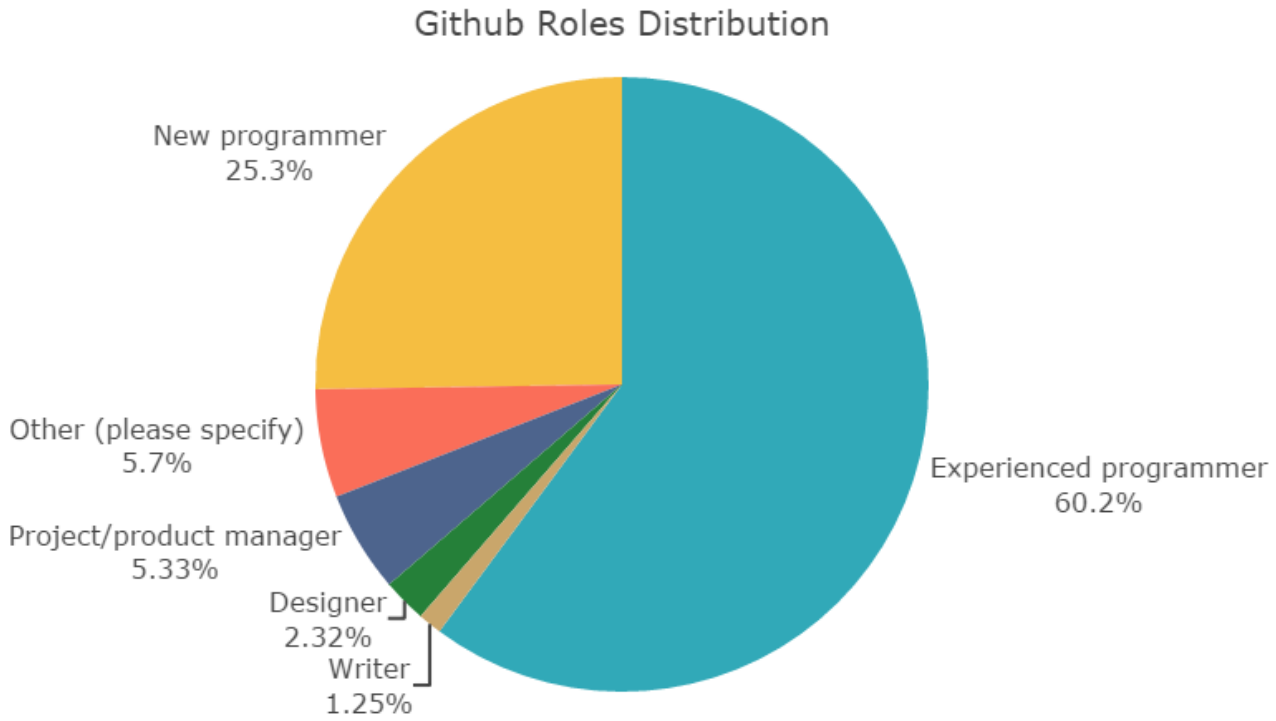


Youngest User: 1.5 days
Oldest User: 2446.5 days (7 yrs)
Avg Account Age: 851.5 days (2 yrs)

- Surveyed on 1600 active account within a 30-day period.
- 67% of accounts are less then 3 years.
- The account age distribution is close to right half of a Gaussian distribution.

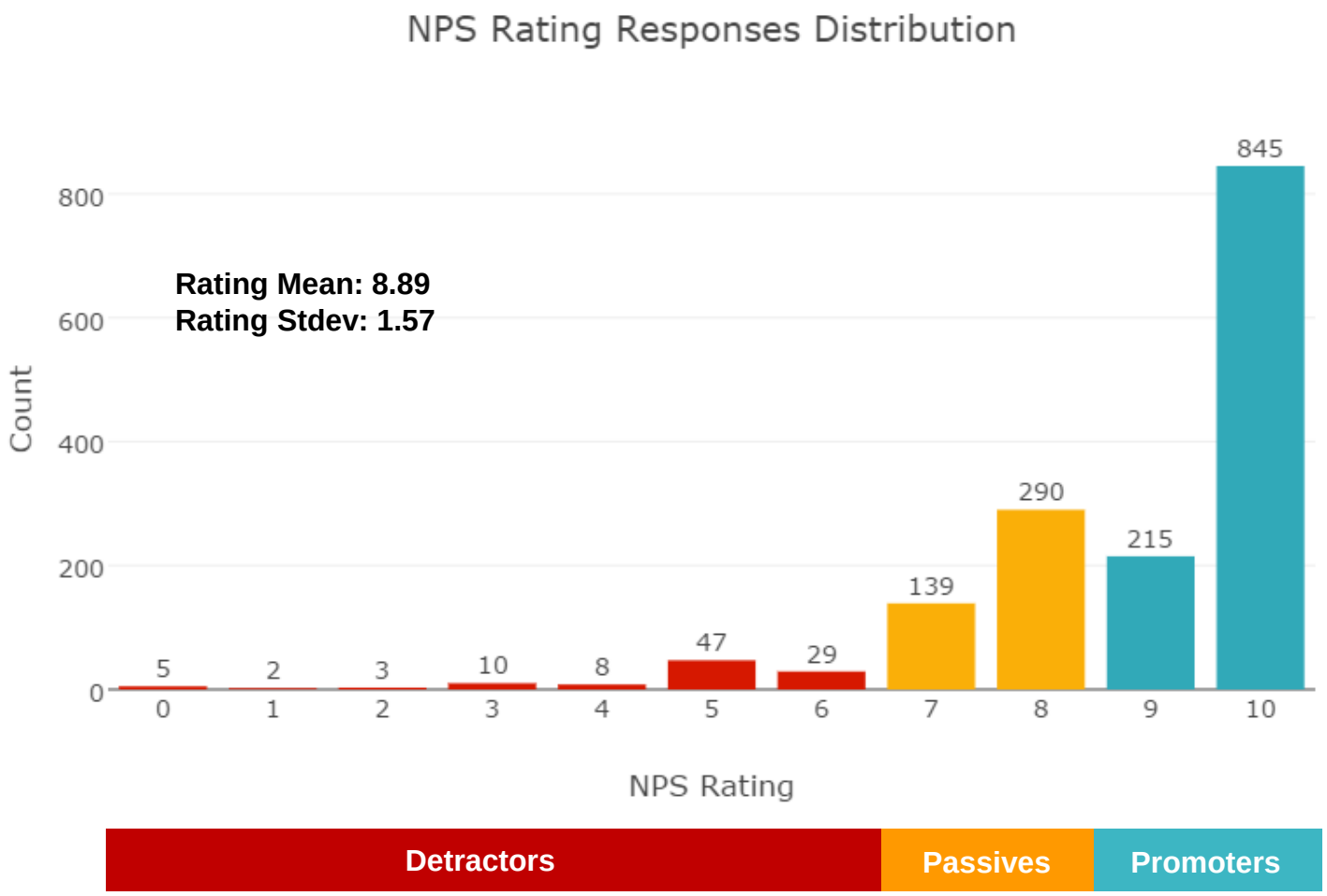
General Survey Info





- Most people consider GitHub as a tool for open source projects.
- 75% of GitHub users (sampled) are programmers (new/experienced).

NPS Analysis



- **NPS = 60.0%**
- **Margin of Error = 1.53%**
- **NPS 95% Confidence Interval : [57%, 63%]**

Total Sample = 1593 (7 of 1600 are Null)
Promoters = 1060
Passives = 429
Detractors = 104

%Promoters = 66.5%
%Passive = 26.9%
%Detractors = 6.6%

NPS Calculation Python Code

NPS = %Promoters -%Detractors

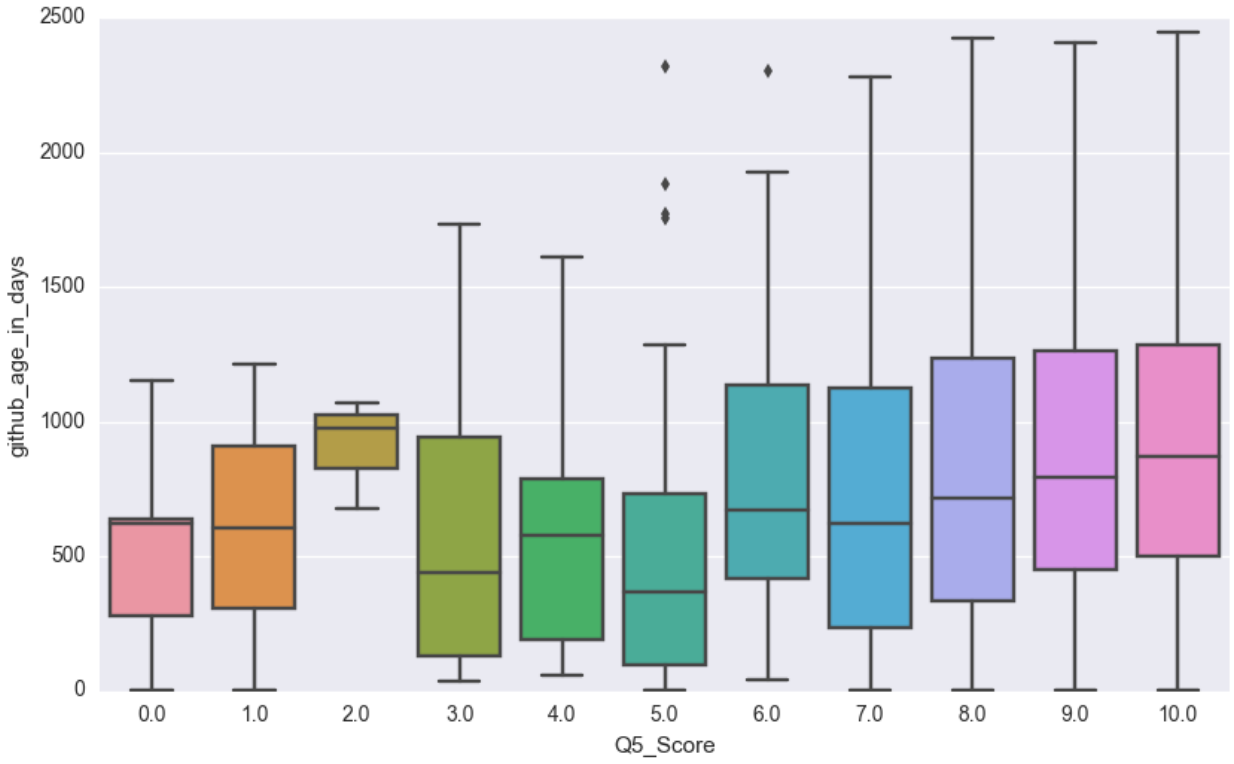
```
#nps_age is the datafile, please see complete code in another attachment file
nps_age = nps_age.dropna(subset=(['Q5_Score']))

#NPS calculation
total_sample = len(nps_age['Q5_Score'])
promoters = len([i for i in list(nps_age['Q5_Score']) if i >= 9])
passives =len([i for i in list(nps_age['Q5_Score']) if i < 9  and i > 6])
detractors= len([i for i in list(nps_age['Q5_Score']) if i <= 6])
prom_percent=float(promoters/total_sample)
pass_percent = float(passives/total_sample)
detr_percent = float(detractors/total_sample)
NPS = prom_percent -detr_percent
```

$$MOE = \frac{\sqrt{(1-NPS)^2(a)+(0-NPS)^2(b)+(-1-NPS)^2(c)}}{\sqrt{n}}$$ (a: %promoters b:%passives c:%detractors)

```
#Margin of Error calculation
Var_NPS = ((1-NPS)**2)*prom_percent + ((0-NPS)**2)*pass_percent + ((-1-NPS)**2)*detr_percent
MOE = ((Var_NPS)**(1/2))/((total_sample)**(1/2))
95% confidence interval = [NPS-1.96*MOE,NPS+1.96*MOE]
```

NPS Score vs. Account Age Analysis

Boxplot of NPS raw ratings vs. Account Age

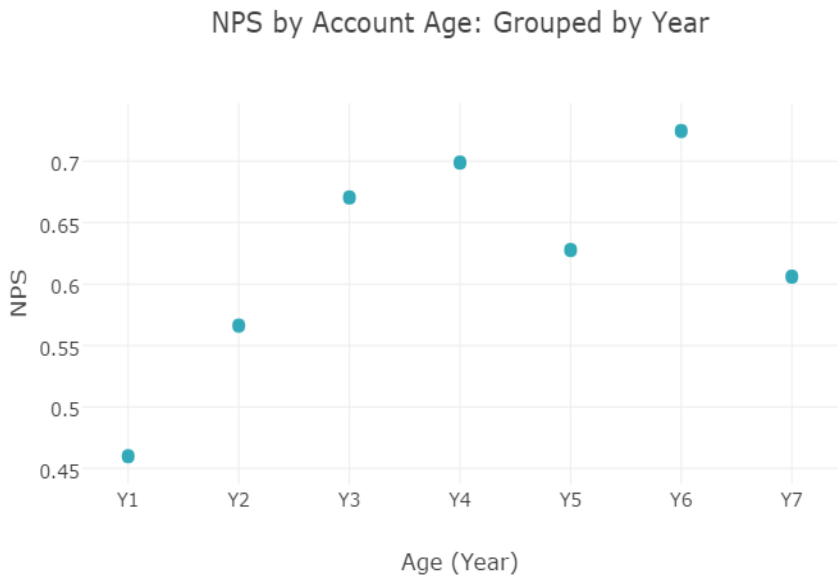


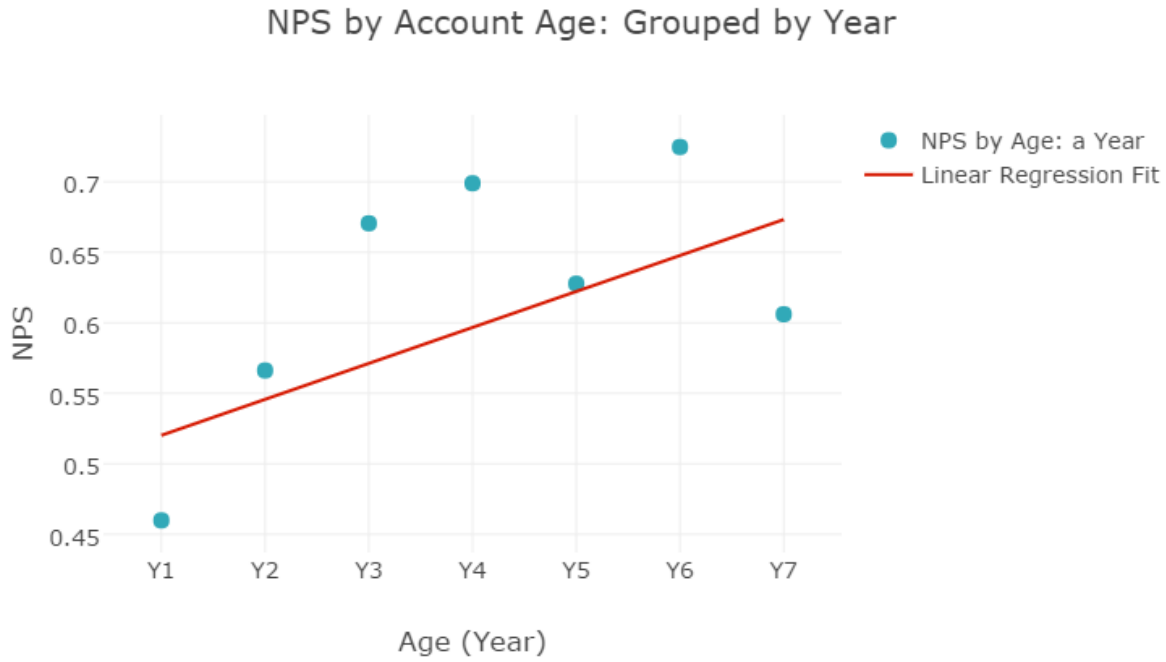
- This boxplot shows that the account age values are widely spread in each rating response (score:0-10); therefore, the variation is big.
- People who gave higher ratings have slightly higher average account age.

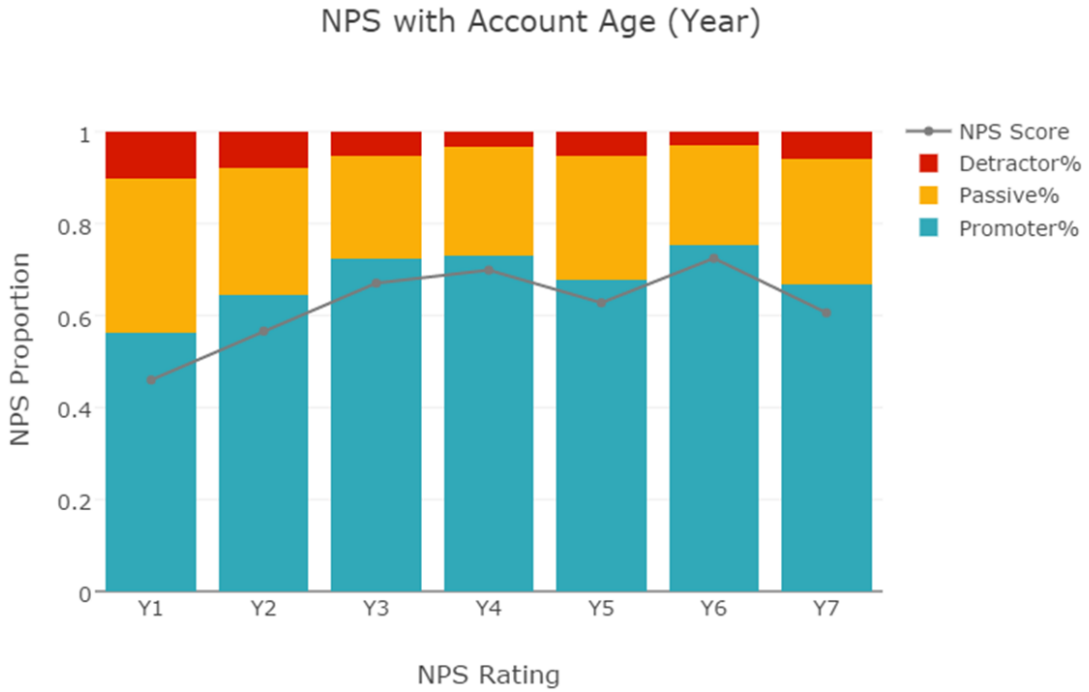
NPS vs. Account Age Regression Model

- Because NPS score is calculated based on a portion of data, the customer accounts in this survey should be categorized into several groups first.
- As the account age is a continuous variable, I grouped the users into year-level buckets by their account age.
- Users are categorized into 7 groups from account age of 1 year to 7 years. Then, I calculated the NPS for each group.
- The scatter plot (NPS vs. account age) reflects that there is some trend that NPS score increases with the age; thus, the two variables could have a linear correlation.
- Hypothesis: older users have higher NPS scores.

Group	NPS	Total Account	Promoter%	Passive %	Detractor %
Y1	46%	374	56%	34%	10%
Y2	57%	355	65%	28%	8%
Y3	67%	346	72%	23%	5%
Y4	70%	279	73%	24%	3%
Y5	63%	137	68%	27%	5%
Y6	72%	69	75%	22%	3%
Y7	61%	33	67%	27%	6%







- A linear regression model is used to fit NPS score and account age.

OLS Regression Results

Dep. Variable:	NPS	R-squared:	0.374
Model:	OLS	Adj. R-squared:	0.249
Method:	Least Squares	F-statistic:	2.988
Date:	Sat, 08 Apr 2017	Prob (F-statistic):	0.144
Time:	19:25:38	Log-Likelihood:	9.1105
No. Observations:	7	AIC:	-14.22
Df Residuals:	5	BIC:	-14.33
Df Model:	1		
Covariance Type:	nonrobust		

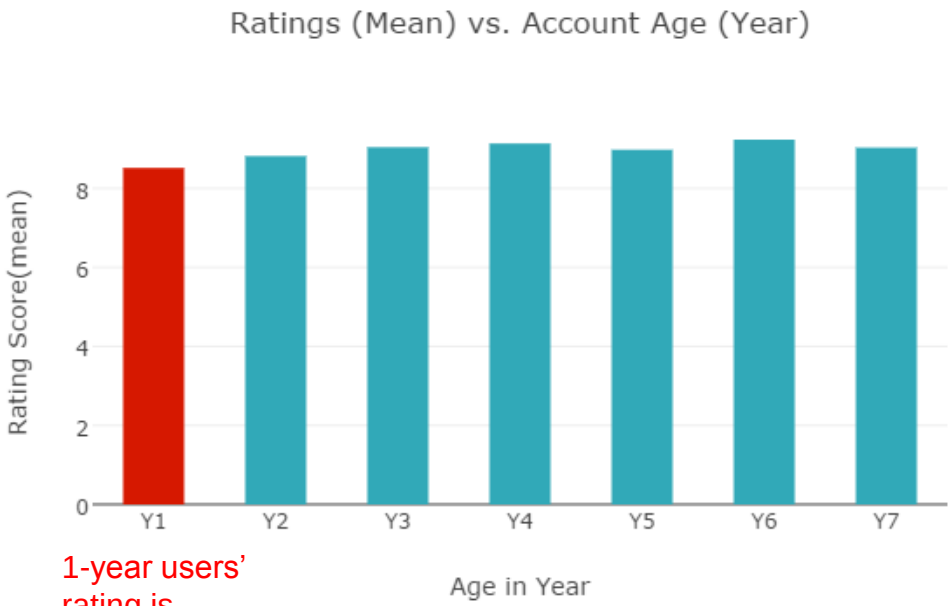
	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
Intercept	0.5202	0.066	7.900	0.001	0.351	0.689
Age_Year_num	0.0255	0.015	1.729	0.144	-0.012	0.063

- The regression model indicates that there is a partial positive correlation between NPS and the account age, especially in the first few years range.
- NPS scores have a drop with 5-yr and 7-yr accounts due to decreased promoters. But the sample sizes of 5 – 7yr accounts are much smaller than 1-4yr. Thus, the results may be biased.
- However, the model is not very strong since R-squared value is 0.374 (1 means a perfect fit). This suggests that we would need more data points to demonstrate the hypothesis.

ANOVA & Post Hoc Test on Raw Rating Reponses

OLS Regression Results

Dep. Variable:	Q5_Score	R-squared:	0.023
Model:	OLS	Adj. R-squared:	0.019
Method:	Least Squares	F-statistic:	6.114
Date:	Sat, 08 Apr 2017	Prob (F-statistic):	2.40e-06
Time:	19:25:46	Log-Likelihood:	-2960.1
No. Observations:	1593	AIC:	5934.
Df Residuals:	1586	BIC:	5972.
Df Model:	6		
Covariance Type:	nonrobust		

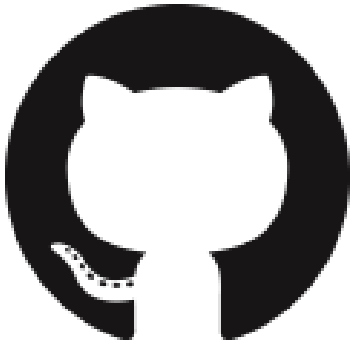


1-year users' rating is significantly lower than others

- Since we only have 1 calculated NPS for each account age group, we can not perform statistical tests to compare the NPS difference among groups.
- I used the raw recommendation rating responses for each age groups to get idea of if there is significant difference among scores.
- ANOVA test (compares the mean scores among groups) indicates that there is a significant difference.
- Post-Hoc Tukey HSD test shows that only 1-yr account group has a score much lower from others.

Suggestions from NPS vs. Age Analysis

- We can categorized the users into more groups based on the age to get more NPS data points. Then, we may have a better-fit linear regression model.
- Also, we would need more survey results from old users (5-7 years) to make the results more convincible.

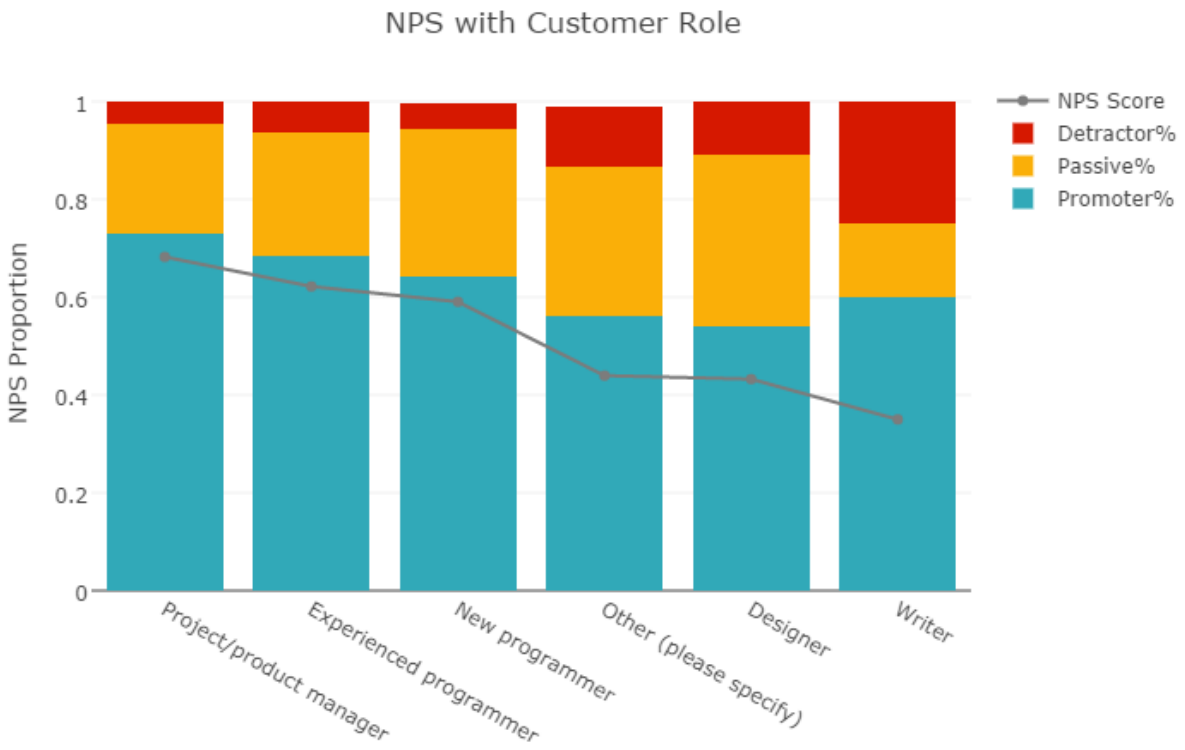


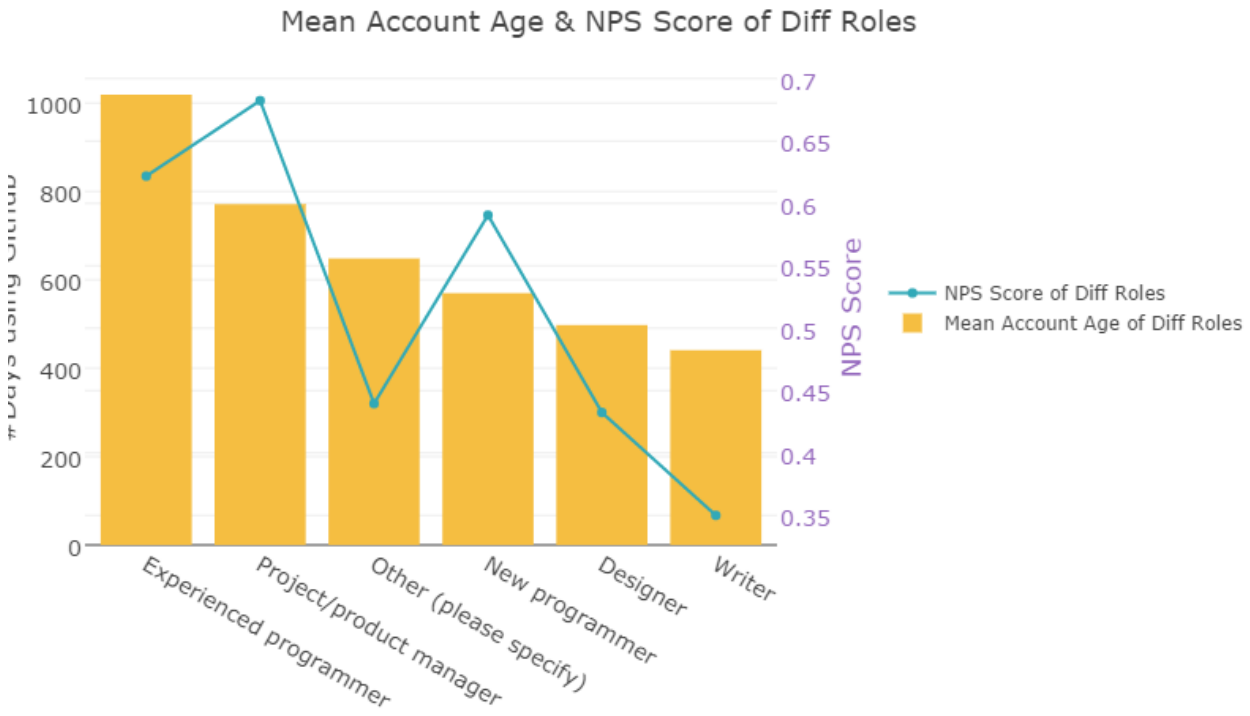
- Even the model is not accurate enough at this point, it provides us some insights.
- New users have a much lower NPS score indicates that they have some barriers at the beginning to use/like GitHub (such as technical issues).
- We can segment the new users and do a deeper analysis on this group to understand their needs.

- Besides the account age, we can include more variables (such as user role, free or paid account) to create a multivariable regression model or a classification model to predict NPS score.

- Since rating drops in old user groups, we should investigate on their issues to satisfy their demands, and to retain them in the long run.

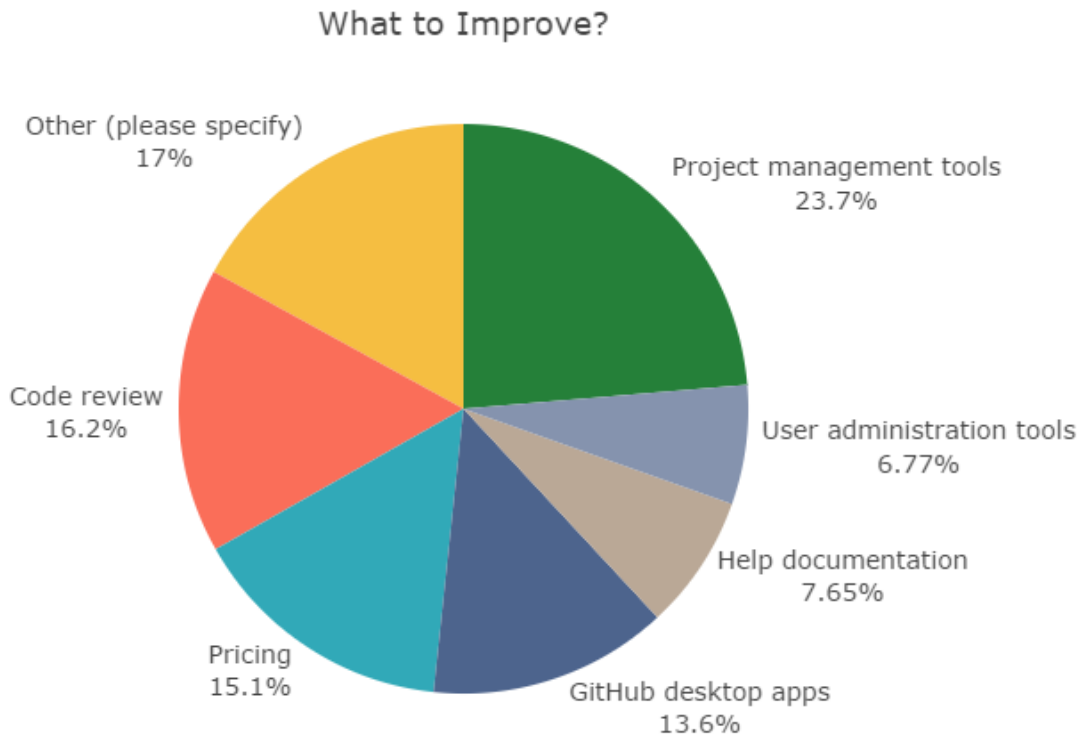
User Role Analysis





- Segment the survey users into 6 groups based on their roles. NPS was calculated for each group.
- The majority of GitHub accounts are programmers and project/product managers (total of 90%, shown in the previous slide). They also gave a positive review (higher NPS within the groups, left figure above), and they used GitHub for a relatively longer time (right figure). Therefore, project/product managers and programmers can be considered as the most valuable customers for GitHub in the long run.
- Customers with non-technical roles (such as designers and writers) have lower NPS scores. Similar to new users group (previous analysis), I'm assuming that they may face some challenges when using GitHub (most likely relates to technical side). Or GitHub may not be a very handy tool for their jobs compared to programmers.

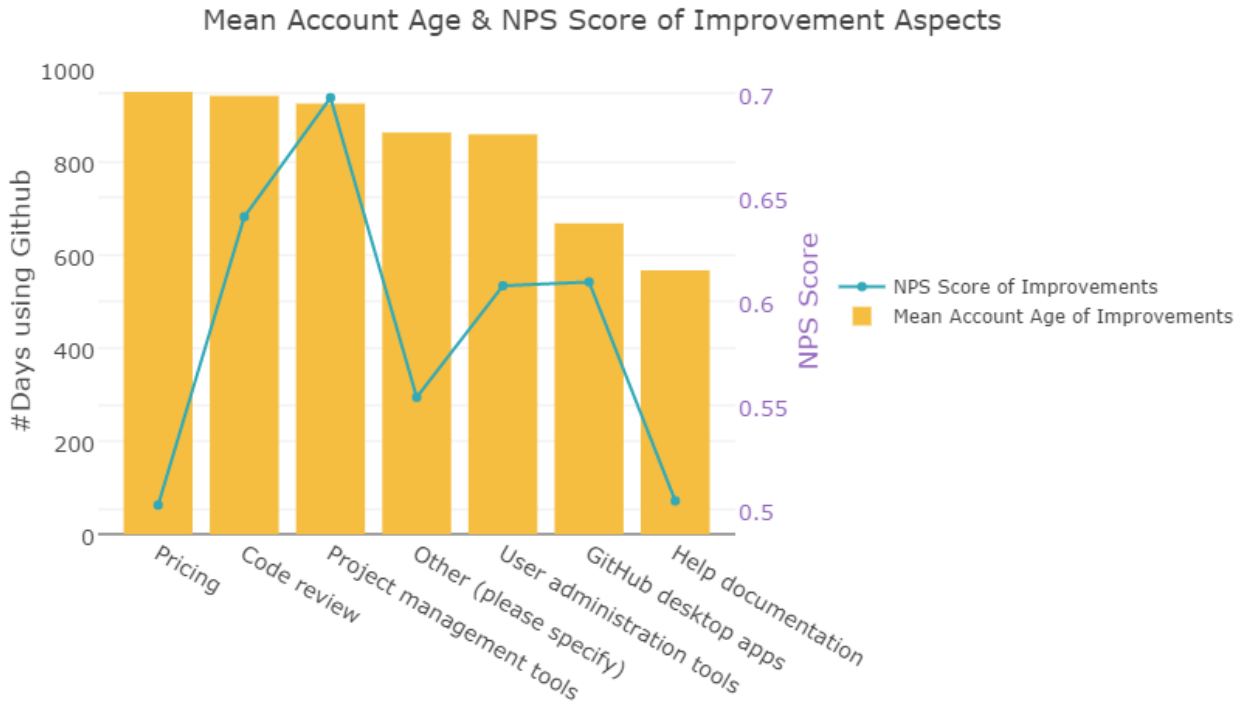
Improvement Questions Analysis

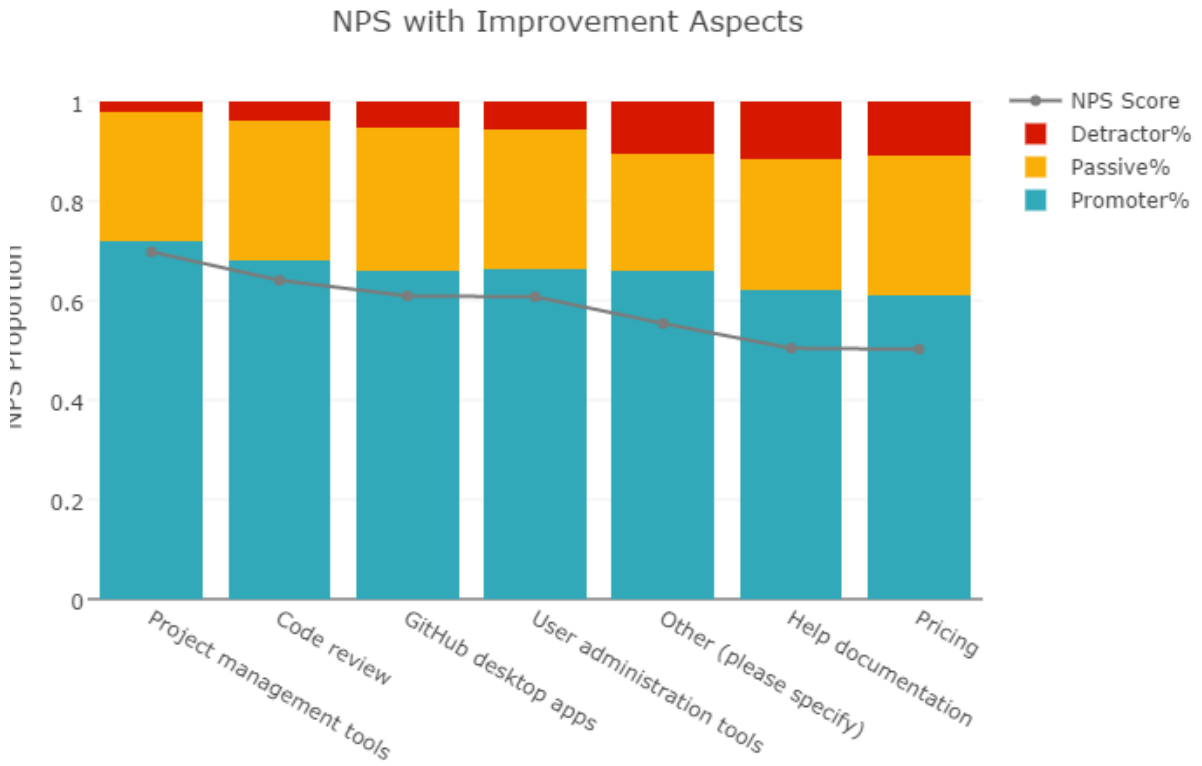


Improvements	NPS	Total Users	Promoter%	Passive%	Detractor %
Project management tools	70%	374	72%	26%	2%
Code review	64%	256	68%	28%	4%
GitHub desktop apps	61%	215	66%	29%	5%
User administration tools	61%	107	66%	28%	6%
Other (please specify)	55%	269	66%	24%	10%
Help documentation	50%	121	62%	26%	12%
Pricing	50%	239	61%	28%	11%

- Project management tools, code review and pricing are the three main aspects that users want GitHub to improve. User groups that voted for project management/code review actually have a high NPS (70% and 64%). This may imply they like the product overall but there are some functions can be better.

- Many users (239) ask for an improvement in pricing. However, their NPS is pretty low (50%), which indicates that pricing would be a potential important task for GitHub. It may be a barrier for a customer to continue using Git.





- Segment users into 7 groups based on their choice of improvements. NPS was calculated for each group.
- The left figure shows that users(total of 239) who votes for pricing have used GitHub for longest time (avg. of 950 days). However, they have a worst NPS.
- This strongly implicates pricing is a challenge for GitHub to keep old users happy and to have them stick with the community.

NPS vs. Detailed Segmentation

Segment the users by their account age (in year level) and choice of improvements. For example (Year1_Project management, Year1_Pricing, etc.)

The segmentations with lowest NPS are listed as below.

Groups	NPS	Total Users
7 Year – Pricing	17%	6
1 Year – Other	28%	63
6 Year – Pricing	33%	12
6 Year – User Admin Tools	33%	6
2 Year – Help Documents	35%	34
5 Year – Pricing	38%	26
1 Year – Pricing	42%	38
2 Year – Pricing	43%	56

- Old users and new users (account age of 1,2,5,6,7 years) both wish the pricing can be improved, especially for people who used GitHub for a long time. Although the group sample size is not very big, improve pricing may effectively increase NPS in the future.

- New users (2 years) who want an improvement in help documentations have a low NPS. I'm guessing they may need more help to overcome technical issues at the beginning .

NPS vs. Detailed Segmentation

Segment the users by their role and choice of improvements. For example (Designer_Project management, Experienced Programmer_Pricing, etc.)

The segmentations with low NPS are listed as below.

Groups	NPS	Total Users
Project/product manager_Help documentation	33%	9
Designer_GitHub desktop apps	42%	12
Experienced programmer_Pricing	51%	167
New programmer_Help documentation	56%	55
Experienced programmer_Other	51%	170
Project/product manager_Pricing	57%	7
New programmer_Pricing	57%	50
New programmer_Project management tools	57%	81

- A significant amount of experienced programmers who complained pricing gave a low NPS. Again, it implies pricing is an important task to address. It also could be a chance to improve overall NPS.
- Non-programmers who complained help documentation and desktop apps as problems also resulted in a low NPS. This suggests that GitHub could develop a more user-friendly environment for non-technical people.

Improvements/Changes Text Analysis

- Since several survey questions are text-based, it would be very time-consuming to read all results.
- I performed a natural language processing technique: Latent Semantic Analysis (LSA) by python packages to get an idea of top concepts of all text-based recommendations in the survey.
- The followings are the top concepts generated by LSA.

Concept 1:	Concept 2:	Concept 3:	Concept 4:	Concept 5:
linux	free	github	really	review
make	private	tools	free	code
app	free private	way	much	project
great	private	good	free private	windows
better	repositories	make	repositories	repo
pull	repositories	apps linux	project	issue
use	free private	private	management	branch
interface	repositories	new	users	users
pricing	repo	github apps	code	app
tools	repos	linux	also	need
	private repos		would	
	better		windows	

The main story behind:

Improve GitHub Linux/Windows App!

Provide Free Private Respositories!

Takeaway & Future Analysis

Key Points:

1. Overall 60% NPS score.
2. NPS does increase with the account age; however, there is a drop for old users (5, 7 years). This may be due to they are not happy with the pricing.
3. New users have a low NPS, which may be caused by some technical entry barriers (need better apps or help documentations).
4. Among with other improvements, pricing is a major complain for almost all level users; it may have a significant role in leading to a lower NPS.
5. A major portion of users want free private repository and better linux/windows apps.

Future Work:

1. Segment the accounts into detailed groups for in-depth analysis: such as new users, old users, programmers, non-tech users. So we can understand how different type of users feel about GitHub and their demands.
2. Investigate in whether pricing is an issue for people to give low scores. This could turn into an effective way to increase NPS.
3. Improve GitHub products (project management, code review, linux/windows apps) to make users happy.
4. Identify which user group bring most value/revenue. Understand their needs to retain high-value customers.